Number of Digits

```cpp
int numDigits( int number );

int main()
{
   srand( static_cast< int >( time( 0 ) ) );

   int number = rand();

   cout << "The number of digits of " << number << " is ";

   cout << numDigits( number ) << endl << endl;
}

int numDigits( int number )
{
   int count = 0;
   while( number > 0 )
   {
      count += 1;
      number /= 10;
   }
   return count;
}
```

```
int numDigits( int number )
{
   if( number < 10 )
      return 1;

   int count = 0;
   while( number / 10 > 0 )
   {
      count += 1;
      number /= 10;
   }
   return 1 + count;
}

int numDigits( int number )
{
   int count = 0;
   while( number > 0 )
   {
      count += 1;
      number /= 10;
   }
   return count;
}
```

```
int numDigits( int number )
{
   if( number < 10 )
      return 1;

   int count = 0;
   while( number / 10 > 0 )
   {
      count += 1;
      number /= 10;
   }
   return 1 + count;
}

int numDigits( int number )
{




}
```

```
int numDigits( int number )
{
   if( number < 10 )
      return 1;

   int count = 0;
   while( number / 10 > 0 )
   {
      count += 1;
      number /= 10;
   }
   return 1 + count;
}

int numDigits( int number )
{
   if( number < 10 )
      return 1;

   return                             ;
}
```

```
int numDigits( int number )
{
   if( number < 10 )
      return 1;

   int count = 0;
   while( number / 10 > 0 )
   {
      count += 1;
      number /= 10;
   }
   return 1 + count;
}

int numDigits( int number )
{
   if( number < 10 )
      return 1;

   return 1 + numDigits( number / 10 );
}
```

```
int main()
{
   srand( static_cast< int >( time( 0 ) ) );

   int number = rand();

   cout << "The number of digits of " << number << " is ";

   cout << numDigits( number ) << endl << endl;
}                        7




                         7
   int numDigits( int number )
   {
      if( number < 10 )
         return 1;

      return 1 + numDigits( number / 10 );
   }
```

```
int main()
{
   srand( static_cast< int >( time( 0 ) ) );

   int number = rand();

   cout << "The number of digits of " << number << " is ";

   cout << numDigits( number ) << endl << endl;
}                       78



                        78
   int numDigits( int number )
   {
      if( number < 10 )
         return 1;
                                 7
      return 1 + numDigits( number / 10 );
   }
```

```
int main()
{
   srand( static_cast< int >( time( 0 ) ) );

   int number = rand();

   cout << "The number of digits of " << number << " is ";

   cout << numDigits( number ) << endl << endl;
}                       789
```

```
                        789
   int numDigits( int number )
   {
      if( number < 10 )
         return 1;
                                78
      return 1 + numDigits( number / 10 );
   }
```

```
int main()
{
   int number = 789;
   cout << numDigits( number ) << endl;
}
```

789

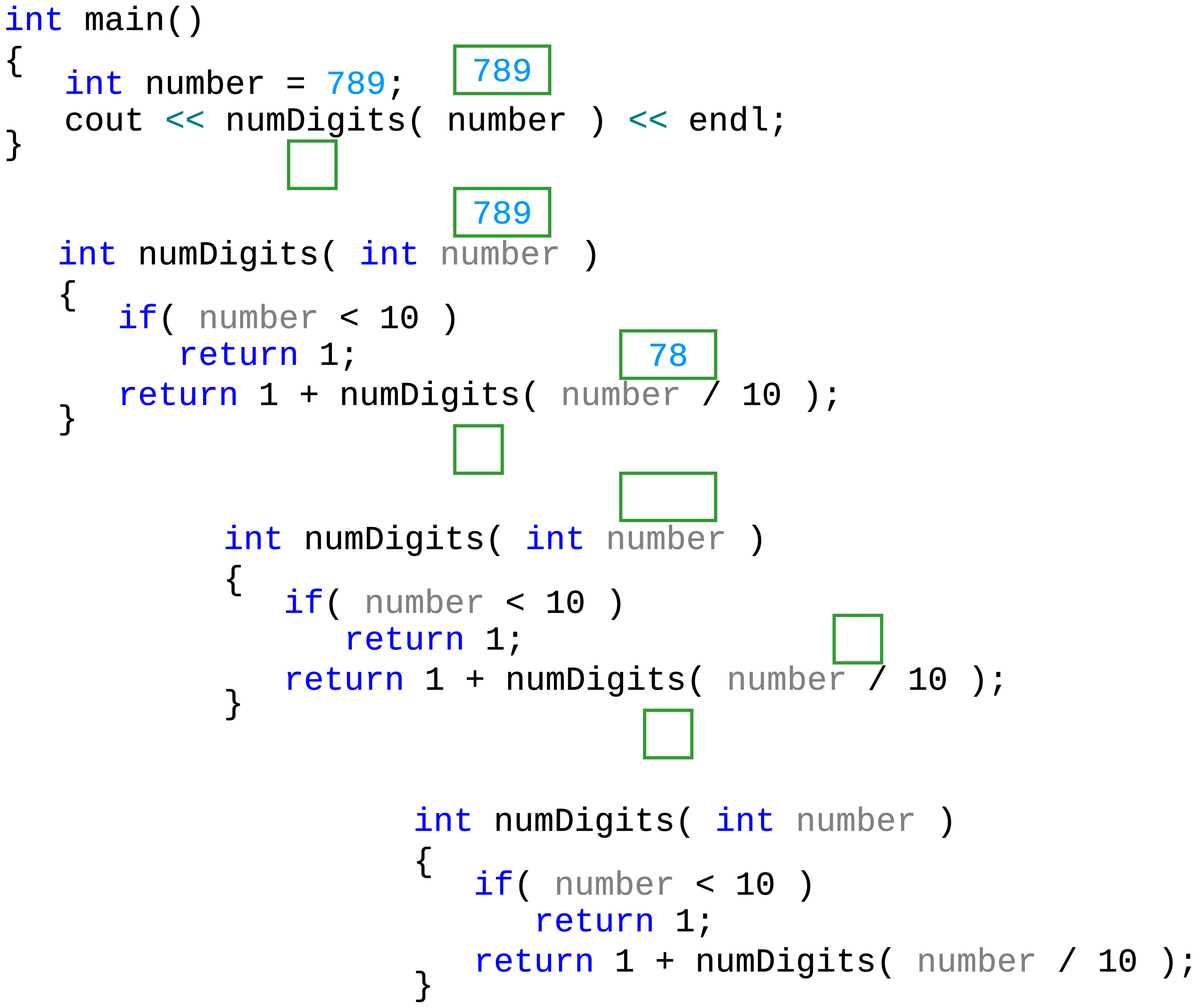
```
int numDigits( int number )
{
   if( number < 10 )
      return 1;
   return 1 + numDigits( number / 10 );
}
```

```
int numDigits( int number )
{
   if( number < 10 )
      return 1;
   return 1 + numDigits( number / 10 );
}
```

```
int numDigits( int number )
{
   if( number < 10 )
      return 1;
   return 1 + numDigits( number / 10 );
}
```

```
int main()
{   int number = 789;     789
    cout << numDigits( number ) << endl;
}


 int numDigits( int number )
 {   if( number < 10 )
        return 1;
     return 1 + numDigits( number / 10 );
 }


     int numDigits( int number )
     {   if( number < 10 )
            return 1;
         return 1 + numDigits( number / 10 );
     }


          int numDigits( int number )
          {   if( number < 10 )
                 return 1;
              return 1 + numDigits( number / 10 );
          }
```

```
int main()
{                      789
   int number = 789;
   cout << numDigits( number ) << endl;
}
              □
                       789
   int numDigits( int number )
   {
      if( number < 10 )
         return 1;                 78
      return 1 + numDigits( number / 10 );
   }
                      □

           int numDigits( int number )
           {
              if( number < 10 )
                 return 1;
              return 1 + numDigits( number / 10 );
           }

                    int numDigits( int number )
                    {
                       if( number < 10 )
                          return 1;
                       return 1 + numDigits( number / 10 );
                    }
```



```
int main()
{
   int number = 789;
   cout << numDigits( number ) << endl;
}
```

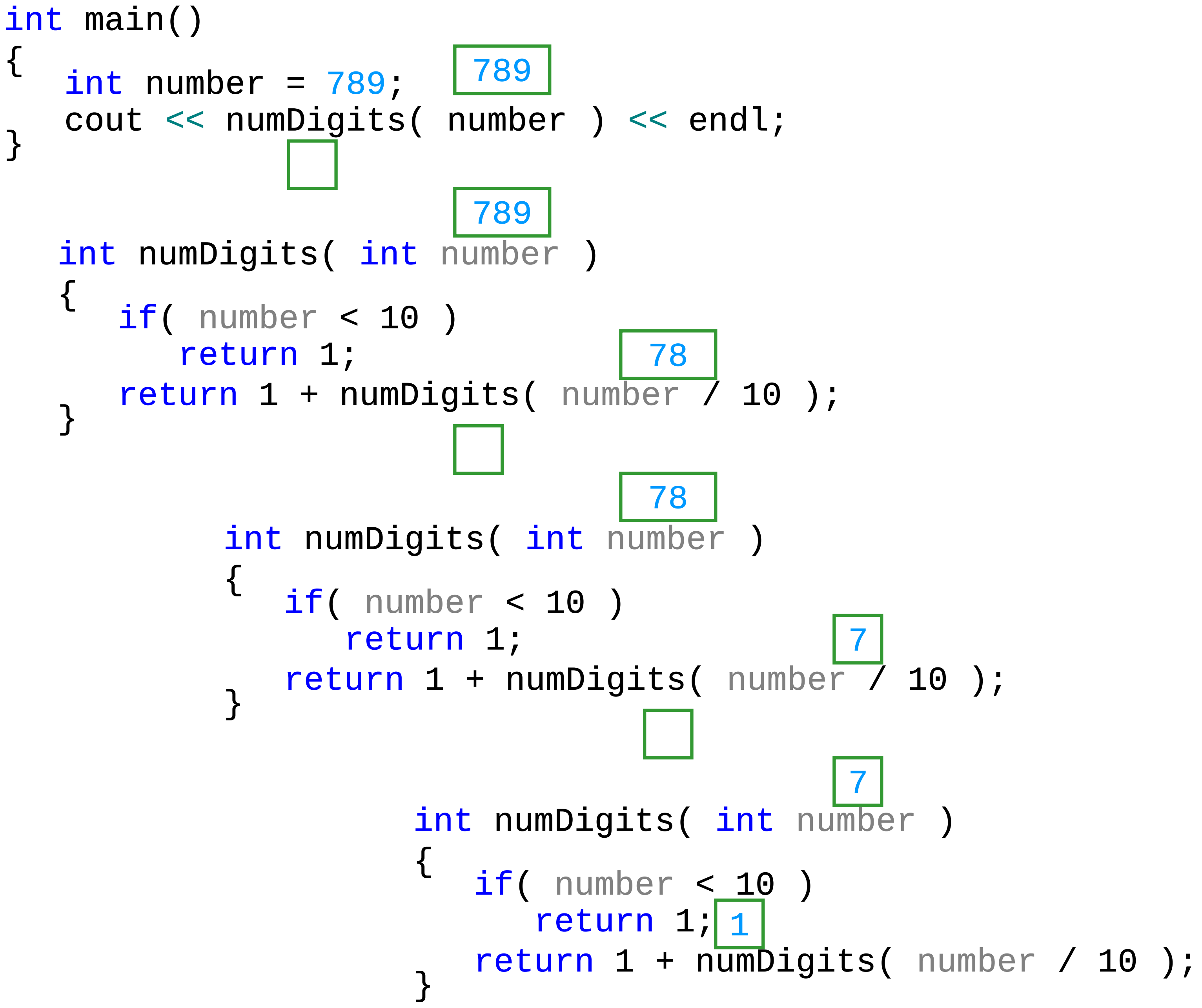
```
int numDigits( int number )
{
   if( number < 10 )
      return 1;
   return 1 + numDigits( number / 10 );
}
```

```
int numDigits( int number )
{
   if( number < 10 )
      return 1;
   return 1 + numDigits( number / 10 );
}
```

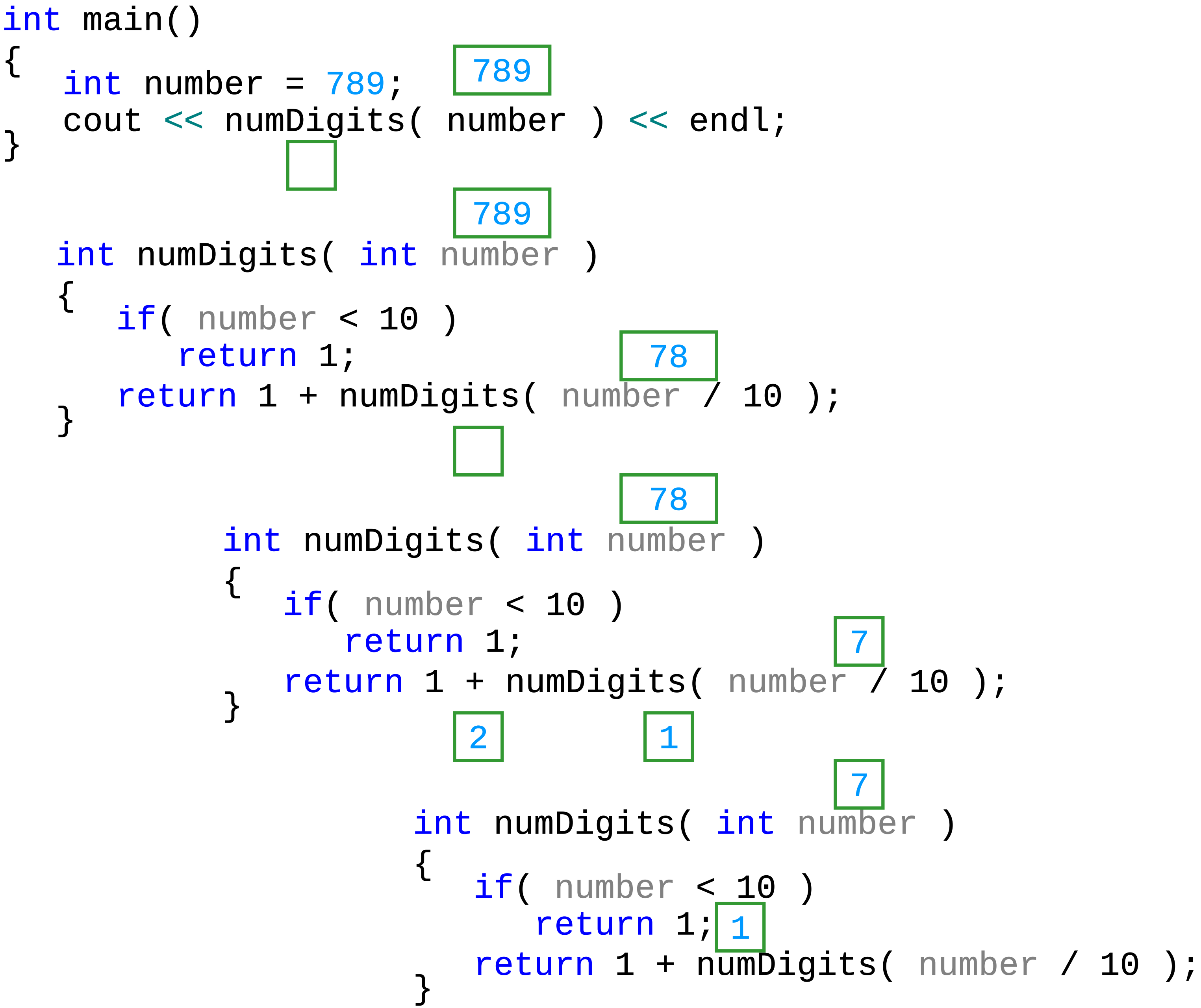
```
int numDigits( int number )
{
   if( number < 10 )
      return 1;
   return 1 + numDigits( number / 10 );
}
```

```
int main()
{                       789
   int number = 789;
   cout << numDigits( number ) << endl;
}
              □
                        789
   int numDigits( int number )
   {
      if( number < 10 )
         return 1;               78
      return 1 + numDigits( number / 10 );
   }
                       □
                                 78
           int numDigits( int number )
           {
              if( number < 10 )
                 return 1;                 7
              return 1 + numDigits( number / 10 );
           }
                                □

                     int numDigits( int number )
                     {
                        if( number < 10 )
                           return 1;
                        return 1 + numDigits( number / 10 );
                     }
```

```
int main()
{                      789
   int number = 789;
   cout << numDigits( number ) << endl;
}
              □
                       789

   int numDigits( int number )
   {
      if( number < 10 )
         return 1;             78
      return 1 + numDigits( number / 10 );
   }
                      □
                               78

           int numDigits( int number )
           {
              if( number < 10 )
                 return 1;               7
              return 1 + numDigits( number / 10 );
           }
                                □
                                         □

                    int numDigits( int number )
                    {
                       if( number < 10 )
                          return 1;□
                       return 1 + numDigits( number / 10 );
                    }
```



```
int main()
                        789
{  int number = 789;
   cout << numDigits( number ) << endl;
}
              □
                        789
   int numDigits( int number )
   {  if( number < 10 )
         return 1;
                                 78
      return 1 + numDigits( number / 10 );
   }
                       □
                                 78
           int numDigits( int number )
           {  if( number < 10 )
                 return 1;
                                           7
              return 1 + numDigits( number / 10 );
           }
                                □
                                           7
                     int numDigits( int number )
                     {  if( number < 10 )
                           return 1; 1
                        return 1 + numDigits( number / 10 );
                     }
```



```
int main()
{
   int number = 789;
   cout << numDigits( number ) << endl;
}
```

```
int numDigits( int number )
{
   if( number < 10 )
      return 1;
   return 1 + numDigits( number / 10 );
}
```

```
int numDigits( int number )
{
   if( number < 10 )
      return 1;
   return 1 + numDigits( number / 10 );
}
```

```
int numDigits( int number )
{
   if( number < 10 )
      return 1;
   return 1 + numDigits( number / 10 );
}
```

```
int main()
{
   int number = 789;   789
   cout << numDigits( number ) << endl;
}
              □
                        789
   int numDigits( int number )
   {
      if( number < 10 )
         return 1;                 78
      return 1 + numDigits( number / 10 );
   }
              3        2
                                 78
           int numDigits( int number )
           {
              if( number < 10 )
                 return 1;                 7
              return 1 + numDigits( number / 10 );
           }
                       2         1
                                           7
                     int numDigits( int number )
                     {
                        if( number < 10 )
                           return 1; 1
                        return 1 + numDigits( number / 10 );
                     }
```

```
int main()
{
   int number = 789;      789
   cout << numDigits( number ) << endl;
}
               3
                       789
   int numDigits( int number )
   {
      if( number < 10 )
         return 1;                78
      return 1 + numDigits( number / 10 );
   }
               3        2
                                78
           int numDigits( int number )
           {
              if( number < 10 )
                 return 1;                  7
              return 1 + numDigits( number / 10 );
           }
                        2         1
                                          7
                    int numDigits( int number )
                    {
                       if( number < 10 )
                          return 1; 1
                       return 1 + numDigits( number / 10 );
                    }
```

```
int numDigits( int number )
{
   if( number < 10 )
      return 1;

   return 1 + numDigits( number / 10 );
}
```

```
int numDigits( int number )
{
   if( number > 9 )
      return 1 + numDigits( number / 10 );

   return 1;
}
```

```
int main()
{
   int n = 789;
   cout << f( n ) << endl;
}
              789
```

```
        789
int f( int n )
{
   if( n < 10 ) return 1;
   return 1 + f( n / 10 );
}
                 78

        78
int f( int n )
{
   if( n < 10 ) return 1;
   return 1 + f( n / 10 );
}
                 7

        7
```

```
        789
int f( int n )
{
   if( n > 9 ) return 1 + f( n / 10 );
   return 1;                    78
}

        78
int f( int n )
{
   if( n > 9 ) return 1 + f( n / 10 );
   return 1;                     7
}

        7
```

Summing Digits

```
int sumDigits( int number );

int main()
{
   srand( static_cast< int >( time( 0 ) ) );

   int number = rand();

   cout << "The sum of digits of " << number << " is ";

   cout << sumDigits( number ) << endl << endl;
}

int sumDigits( int number )
{
   int sum = 0;
   while( number > 0 )
   {
      sum += number % 10;
      number /= 10;
   }
   return sum;
}
```

```
int sumDigits( int number );

int main()
{
   srand( static_cast< int >( time( 0 ) ) );

   int number = rand();

   cout << "The sum of digits of " << number << " is ";

   cout << sumDigits( number ) << endl << endl;
}

int sumDigits( int number )
{




}
```

```
int sumDigits( int number );

int main()
{
   srand( static_cast< int >( time( 0 ) ) );

   int number = rand();

   cout << "The sum of digits of " << number << " is ";

   cout << sumDigits( number ) << endl << endl;
}

int sumDigits( int number )
{
   if( number < 10 )
      return number;

   return sumDigits(             ) +             ;
}
```

```cpp
int sumDigits( int number );

int main()
{
   srand( static_cast< int >( time( 0 ) ) );

   int number = rand();

   cout << "The sum of digits of " << number << " is ";

   cout << sumDigits( number ) << endl << endl;
}

int sumDigits( int number )
{
   if( number < 10 )
      return number;

   return sumDigits( number / 10 ) + number % 10;
}
```

```
int main()
{
   int number = 789;   789
   cout << sumDigits( number ) << endl;
}
              24

                       789
   int sumDigits( int number )
   {
      if( number < 10 )
         return number;     78       24      9
      return sumDigits( number / 10 ) + number % 10;
   }
                15

                                78
           int sumDigits( int number )
           {
              if( number < 10 )
                 return number;       7      15      8
              return sumDigits( number / 10 ) + number % 10;
           }
                         7

                                          7
                    int sumDigits( int number )
                    {
                       if( number < 10 )
                          return number;  7
                       return sumDigits( number / 10 ) + number % 10;
                    }
```

```
int sumDigits( int number )
{
   if( number < 10 )
      return number;

   return sumDigits( number / 10 ) + number % 10;
}
```

```
int sumDigits( int number )
{
   if( number > 9 )
      return sumDigits( number / 10 ) + number % 10;

   return number;
}
```

```
int main()
{
    int n = 789;
    cout << f( n ) << endl;
}
               24
```

```
          789
int f( int n )
{
    if( n < 10 ) return n;
    return f( n / 10 ) + n % 10;
}
             78     15    9
```

```
          78
int f( int n )
{
    if( n < 10 ) return n;
    return f( n / 10 ) + n % 10;
}
              7           8
```

```
          7
```

```
          789
int f( int n )
{
    if( n > 9 ) return f( n / 10 ) + n % 10;
    return n;
}
                         78     15    9
```

```
          78
int f( int n )
{
    if( n > 9 ) return f( n / 10 ) + n % 10;
    return n;
}
                          7           8
```

```
          7
```

Prints Digits in the Reverse Order

```
void reverse( int number );

int main()
{
   int number;
   cin >> number;
   reverse( number );
}

void reverse( int number )
{
   while( number > 0 )
   {
      cout << number % 10;
      number /= 10;
   }
}
```

```
void reverse( int number );

int main()
{
   int number;
   cin >> number;
   reverse( number );
}

void reverse( int number )
{



}
```

```
void reverse( int number );

int main()
{
   int number;
   cin >> number;
   reverse( number );
}

void reverse( int number )
{
   cout <<            ;
   if( number > 9 )
      reverse(             );
}
```

```
void reverse( int number );

int main()
{
   int number;
   cin >> number;
   reverse( number );
}

void reverse( int number )
{
   cout << number % 10;
   if( number > 9 )
      reverse( number / 10 );
}
```

```
int main()
{
   int number = 789;
   reverse( number );
}
```

Output

```
void reverse( int number )
{
   cout << number % 10;
   if( number > 9 ) reverse( number/10 );
}
```

```
             void reverse( int number )
             {
                cout << number % 10;
                if( number > 9 ) reverse( number/10 );
             }
```

```
                          void reverse( int number )
                          {
                             cout << number % 10;
                             if( number > 9 ) reverse( number/10 );
                          }
```

```
      int main()
      {
         int number = 789;
         reverse( number );
      }
```

789

Output

```
void reverse( int number )
{
   cout << number % 10;
   if( number > 9 ) reverse( number/10 );
}

             void reverse( int number )
             {
                cout << number % 10;
                if( number > 9 ) reverse( number/10 );
             }

                          void reverse( int number )
                          {
                             cout << number % 10;
                             if( number > 9 ) reverse( number/10 );
                          }
```

```
int main()
{
   int number = 789;
   reverse( number );
}           789
```

```
9
```

Output

```
                789
void reverse( int number )
{
   cout << number % 10;
   if( number > 9 ) reverse( number / 10 );
}
```

```
             void reverse( int number )
             {
                cout << number % 10;
                if( number > 9 ) reverse( number / 10 );
             }
```

```
                          void reverse( int number )
                          {
                             cout << number % 10;
                             if( number > 9 ) reverse( number / 10 );
                          }
```

```
     int main()
     {
        int number = 789;
        reverse( number );
     }           789

                 789
void reverse( int number )
{
   cout << number % 10;         78
   if( number > 9 ) reverse( number / 10 );
}

             void reverse( int number )
             {
                cout << number % 10;
                if( number > 9 ) reverse( number / 10 );
             }

                          void reverse( int number )
                          {
                             cout << number % 10;
                             if( number > 9 ) reverse( number / 10 );
                          }
```

```
9
```

Output

```
      int main()
      {
         int number = 789;
         reverse( number );
      }            789

                   789
void reverse( int number )
{
   cout << number % 10;            78
   if( number > 9 ) reverse( number / 10 );
}

                                  78
             void reverse( int number )
             {
                cout << number % 10;
                if( number > 9 ) reverse( number / 10 );
             }


                          void reverse( int number )
                          {
                             cout << number % 10;
                             if( number > 9 ) reverse( number / 10 );
                          }
```

```
98
```

Output

```
      int main()
      {
         int number = 789;
         reverse( number );
      }
                    789

                    789
void reverse( int number )
{
   cout << number % 10;          78
   if( number > 9 ) reverse( number/10 );
}

                                 78
             void reverse( int number )
             {
                cout << number % 10;          7
                if( number > 9 ) reverse( number/10 );
             }


                          void reverse( int number )
                          {
                             cout << number % 10;
                             if( number > 9 ) reverse( number/10 );
                          }
```

```
98
```

Output

```
int main()
{
   int number = 789;
   reverse( number );
}
```

789

987

Output

789

```
void reverse( int number )
{
   cout << number % 10;
   if( number > 9 ) reverse( number / 10 );
}
```

78

78

```
void reverse( int number )
{
   cout << number % 10;
   if( number > 9 ) reverse( number / 10 );
}
```

7

7

```
void reverse( int number )
{
   cout << number % 10;
   if( number > 9 ) reverse( number / 10 );
}
```

Prints Digits of An Integer

```
int main()
{
   srand( static_cast< int >( time( 0 ) ) );

   int number = rand();

   cout << "The digits of " << number << ": ";

   digits( number );

   cout << endl << endl;
}

void digits( int number )
{




}
```

```cpp
int main()
{
   srand( static_cast< int >( time( 0 ) ) );

   int number = rand();

   cout << "The digits of " << number << ": ";

   digits( number );

   cout << endl << endl;
}

void digits( int number )
{
   if( number > 9 )
      digits(             );

   cout <<            ;
}
```

```
int main()
{
   srand( static_cast< int >( time( 0 ) ) );

   int number = rand();

   cout << "The digits of " << number << ": ";

   digits( number );

   cout << endl << endl;
}

void digits( int number )
{
   if( number > 9 )
      digits( number / 10 );

   cout << number % 10;
}
```

```
int main()
{
   int number = 789;
   digits( number );
}
```

789

Output

```
void digits( int number )
{
   if( number > 9 ) digits( number / 10 );
   cout << number % 10;
}
```

```
             void digits( int number )
             {
                if( number > 9 ) digits( number / 10 );
                cout << number % 10;
             }
```

```
                          void digits( int number )
                          {
                             if( number > 9 ) digits( number / 10 );
                             cout << number % 10;
                          }
```

```
int main()
{
   int number = 789;
   digits( number );
}
            789
```

Output

```
            789
void digits( int number )
{
   if( number > 9 ) digits( number / 10 );
   cout << number % 10;               78
}
```

```
void digits( int number )
{
   if( number > 9 ) digits( number / 10 );
   cout << number % 10;
}
```

```
void digits( int number )
{
   if( number > 9 ) digits( number / 10 );
   cout << number % 10;
}
```

```
int main()
{
    int number = 789;
    digits( number );
}
```

789

Output

789

```
void digits( int number )
{
    if( number > 9 ) digits( number/10 );
    cout << number % 10;
}
```

78

```
void digits( int number )
{
    if( number > 9 ) digits( number/10 );
    cout << number % 10;
}
```

```
void digits( int number )
{
    if( number > 9 ) digits( number/10 );
    cout << number % 10;
}
```

```
int main()
{
   int number = 789;
   digits( number );
}
```

789

Output

789

```
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;
}
```

78

```
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;
}
```

```
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;
}
```

```
int main()
{
   int number = 789;
   digits( number );
}
           789
```

Output

```
           789
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;
}
```

```
            78
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;          7
}
```

```
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;
}
```

```
int main()
{
    int number = 789;
    digits( number );
}
               789
```

Output

```
               789
void digits( int number )
{
    if( number > 9 ) digits( number/10 );
    cout << number % 10;
}
```

```
                 78
void digits( int number )
{
    if( number > 9 ) digits( number/10 );
    cout << number % 10;          7
}
```

```
void digits( int number )
{
    if( number > 9 ) digits( number/10 );
    cout << number % 10;
}
```

```
int main()
{
   int number = 789;
   digits( number );
}
             789
```

Output

```
             789
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;
}
```

```
             78
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;
}
```

```
             7
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;
}
```

```
int main()
{
   int number = 789;
   digits( number );
}           789
```

```
7
```

Output

```
                789
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;
}
```

```
                78
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;
}
```

```
                7
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;
}
```

```
int main()
{
   int number = 789;
   digits( number );
}
```

789

78

Output

789

```
void digits( int number )
{
   if( number > 9 ) digits( number / 10 );
   cout << number % 10;
}
```

78

```
void digits( int number )
{
   if( number > 9 ) digits( number / 10 );
   cout << number % 10;
}
```

7

```
void digits( int number )
{
   if( number > 9 ) digits( number / 10 );
   cout << number % 10;
}
```

```
int main()
{
   int number = 789;
   digits( number );
}
           789
```

789

Output

```
           789
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;
}
```

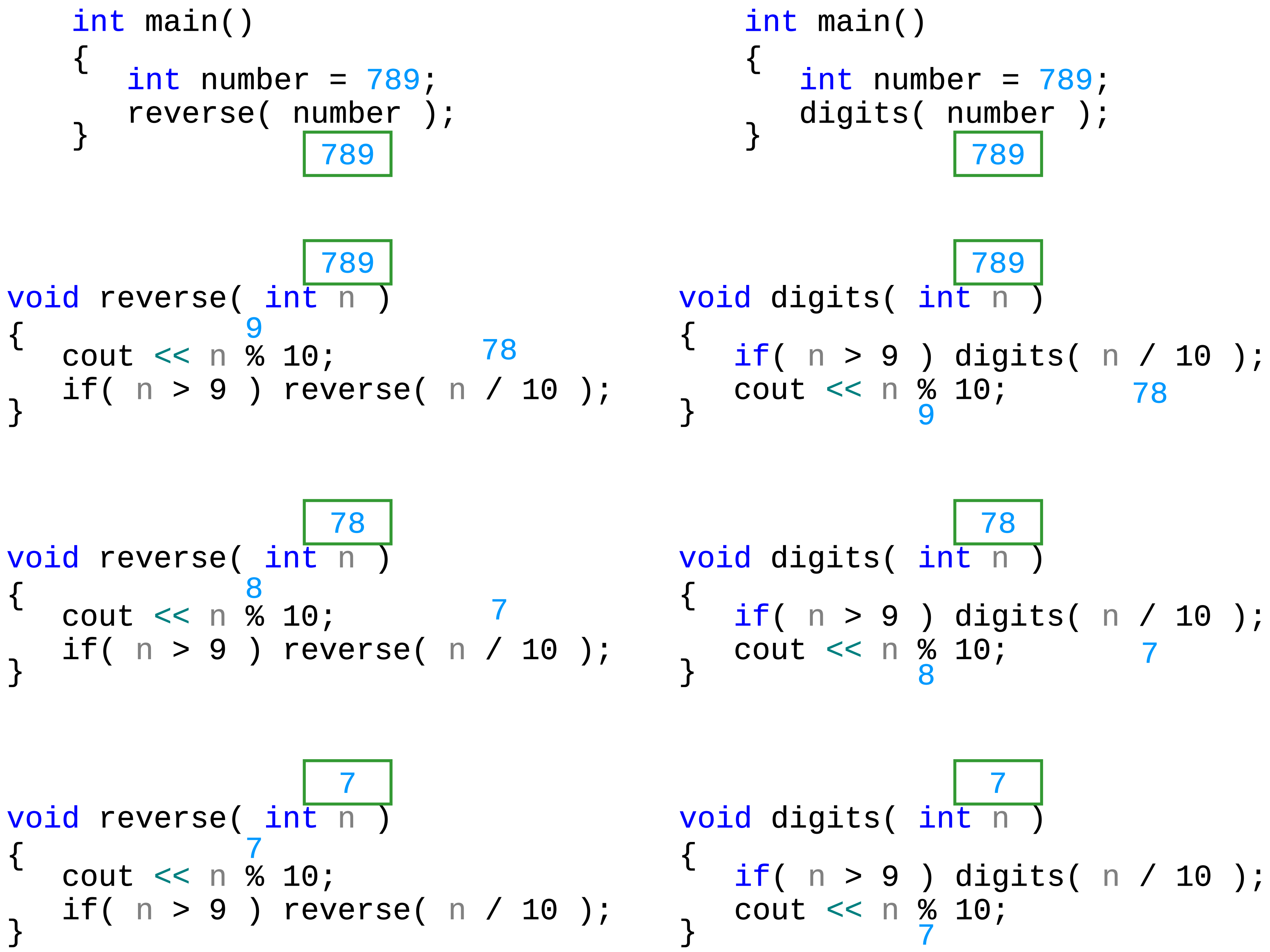
```
                 78
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;
}
```

```
                 7
void digits( int number )
{
   if( number > 9 ) digits( number/10 );
   cout << number % 10;
}
```

Prints Digits of An Integer Forward and Backward

```
void digits( int number )
{
   if( number > 9 )
      digits( number / 10 );

   cout << number % 10;
}

void reverse( int number )
{
   cout << number % 10;

   if( number > 9 )
      reverse( number / 10 );
}
```

```
void digits( int n )
{
   if( n > 9 ) digits( n / 10 );
   cout << n % 10;
}

void reverse( int n )
{
   cout << n % 10;
   if( n > 9 ) reverse( n / 10 );
}
```



```
int main()
{
   int number = 789;
   reverse( number );
}
              789
```

```
                 789
void reverse( int n )
{             9
   cout << n % 10;          78
   if( n > 9 ) reverse( n / 10 );
}
```

```
                 78
void reverse( int n )
{             8
   cout << n % 10;           7
   if( n > 9 ) reverse( n / 10 );
}
```

```
                 7
void reverse( int n )
{             7
   cout << n % 10;
   if( n > 9 ) reverse( n / 10 );
}
```

```
int main()
{
   int number = 789;
   digits( number );
}
            789
```

```
                789
void digits( int n )
{
   if( n > 9 ) digits( n / 10 );
   cout << n % 10;
}                        78
             9
```

```
                78
void digits( int n )
{
   if( n > 9 ) digits( n / 10 );
   cout << n % 10;
}                         7
             8
```

```
                7
void digits( int n )
{
   if( n > 9 ) digits( n / 10 );
   cout << n % 10;
}
             7
```

Recursive Prints an Array

```
  const int arraySize = 3;
  int data[ arraySize ];

  int main()
  {
     for( int i = 0; i < arraySize; i++ )
        data[ i ] = 1 + rand() % 10;

     cout << "\n\nArray values: ";
     recursivePrintArray( arraySize - 1 );
     cout << endl << endl;
  }
```

0	1	2
9	6	7

```
void recursivePrintArray( int last )
{




}
```

```
  const int arraySize = 1;
  int data[ arraySize ];

  int main()
  {
     for( int i = 0; i < arraySize; i++ )
        data[ i ] = 1 + rand() % 10;

     cout << "\n\nArray values: ";
     recursivePrintArray( arraySize - 1 );
     cout << endl << endl;
  }
```

0

0	1	2
9	6	7

```
void recursivePrintArray( int last )
{
   if( last > 0 )
      recursivePrintArray( last - 1 );

   cout << setw( 5 ) << data[ last ];
}
```

```
  const int arraySize = 2;
  int data[ arraySize ];

  int main()
  {
     for( int i = 0; i < arraySize; i++ )
        data[ i ] = 1 + rand() % 10;

     cout << "\n\nArray values: ";
     recursivePrintArray( arraySize - 1 );
     cout << endl << endl;      1
  }



void recursivePrintArray( int last )
{
   if( last > 0 )
                                0
      recursivePrintArray( last - 1 );

   cout << setw( 5 ) << data[ last ];
}
```

0	1	2
9	6	7

```
  const int arraySize = 3;
  int data[ arraySize ];

  int main()
  {
     for( int i = 0; i < arraySize; i++ )
        data[ i ] = 1 + rand() % 10;

     cout << "\n\nArray values: ";
     recursivePrintArray( arraySize - 1 );
     cout << endl << endl;      2
  }


                               □
void recursivePrintArray( int last )
{
   if( last > 0 )
                                1
      recursivePrintArray( last - 1 );

   cout << setw( 5 ) << data[ last ];
}
```

0	1	2
9	6	7

```
int data[ arraySize ];
int main()    3
{
   recursivePrintArray( arraySize - 1 );
}
```

Output

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

0	1	2
9	6	7

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
int data[ arraySize ];
int main()    3
{
   recursivePrintArray( arraySize - 1 );
}                             2
```

Output

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

0	1	2
9	6	7

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
int data[ arraySize ];
int main()     3
{
   recursivePrintArray( arraySize - 1 );
}                              2
```

Output

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

0	1	2
9	6	7

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
int data[ arraySize ];
int main()    3
{
   recursivePrintArray( arraySize - 1 );
}
```

Output

```
                                  2
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];            1
}
```

0	1	2
9	6	7

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
int data[ arraySize ];
int main()     3
{
   recursivePrintArray( arraySize - 1 );
}
```

Output

```
                                   2
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];          1
}
```

```
                                   □
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

0	1	2
9	6	7

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
int data[ arraySize ];
int main()      3
{
   recursivePrintArray( arraySize - 1 );
}
```

Output

```
                              2
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

0	1	2
9	6	7

```
                              1
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];         0
}
```

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
int data[ arraySize ];
int main()    3
{
   recursivePrintArray( arraySize - 1 );
}
```

Output

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
                              1
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];      0
}
```

0	1	2
9	6	7

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
int data[ arraySize ];
int main()     3
{
   recursivePrintArray( arraySize - 1 );
}
```

```
9
```

Output

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

0	1	2
9	6	7

```
                          1
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
                          0
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
int data[ arraySize ];
int main()    3
{
   recursivePrintArray( arraySize - 1 );
}
```

```
9 6
```

Output

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

0	1	2
9	6	7

1

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

0

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
int data[ arraySize ];
int main()    3
{
   recursivePrintArray( arraySize - 1 );
}
```

```
9 6 7
```

Output

```
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

0	1	2
9	6	7

```
                               1
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

```
                               0
void recursivePrintArray( int last )
{
   if( last > 0 ) recursivePrintArray( last - 1 );
   cout << setw( 5 ) << data[ last ];
}
```

Recursive Prints an Array in Reverse Order

```
const int arraySize = 1;
int data[ arraySize ];

int main()
{
   for( int i = 0; i < arraySize; i++ )
      data[ i ] = 1 + rand() % 10;

   cout << "\n\nArray values in reverse order: ";
   recursivePrintReverse( arraySize - 1 );
   cout << endl << endl;
}
```

0	1	2
9	6	7

```
void recursivePrintReverse( int last )
{




}
```

```
const int arraySize = 1;
int data[ arraySize ];

int main()
{
   for( int i = 0; i < arraySize; i++ )
      data[ i ] = 1 + rand() % 10;

   cout << "\n\nArray values in reverse order: ";
   recursivePrintReverse( arraySize - 1 );
   cout << endl << endl;
}
```

0	1	2
9	6	7

```
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];

   if( last > 0 )
      recursivePrintReverse( last - 1 );
}
```

```
 int data[ arraySize ];
 int main()    3
 {
    recursivePrintReverse( arraySize - 1 );
 }                                 2
```

Output

```
                                  2
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}
```

0	1	2
9	6	7

```
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}
```

```
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}
```

```
int data[ arraySize ];
int main()    3
{
   recursivePrintReverse( arraySize - 1 );
}                                 2
```

```
7
```

Output

```
                                  2
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}
```

0	1	2
9	6	7

```
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}
```

```
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}
```

```
  int data[ arraySize ];
  int main()    3
  {
     recursivePrintReverse( arraySize - 1 );
  }                                2
```

```
7
```

Output

```
                                  2
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}                                            1
```

0	1	2
9	6	7

```
                                  1
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}
```

```
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}
```

```
int data[ arraySize ];
int main()      3
{
   recursivePrintReverse( arraySize - 1 );
}                                    2
```

```
   76
```

Output

```
                                     2
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}                                            1
```

0	1	2
9	6	7

```
                                     1
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}
```

```
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}
```

```
int data[ arraySize ];
int main()    3
{
   recursivePrintReverse( arraySize - 1 );
}                                2
```

7 6

Output

```
                                  2
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}                                            1
```

0	1	2
9	6	7

```
                                  1
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}                                            0
```

```
                                  0
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}
```

```
int data[ arraySize ];
int main()    3
{
   recursivePrintReverse( arraySize - 1 );
}                                2
```

```
7 6 9
```

Output

```
                                  2
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}                                            1
```

0	1	2
9	6	7

```
                                  1
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}                                            0
```

```
                                  0
void recursivePrintReverse( int last )
{
   cout << setw( 5 ) << data[ last ];
   if( last > 0 ) recursivePrintReverse( last - 1 );
}
```

Recursive Prints an Array Forward and Backward

```
void recursivePrintArray( int last )
{
   if( last > 0 )
      recursivePrintArray( last - 1 );

   cout << " " << data[ last ];
}

void recursivePrintReverse( int last )
{
   cout << " " << data[ last ];

   if( last > 0 )
      recursivePrintReverse( last - 1 );
}
```

```
void recursivePrintArray1( int first, int last )
{


}

void recursivePrintArray2( int first, int last )
{


}

void recursivePrintReverse1( int first, int last )
{


}

void recursivePrintReverse2( int first, int last )
{


}
```

```
void recursivePrintArray1( int first, int last )
{


}

void recursivePrintArray2( int first, int last )
{
   if( first < last )
      recursivePrintArray2( first, last - 1 );
   cout << " " << data[ last ];
}

void recursivePrintReverse1( int first, int last )
{


}

void recursivePrintReverse2( int first, int last )
{


}
```

```
void recursivePrintArray1( int first, int last )
{
   cout << " " << data[ first ];
   if( first < last )
      recursivePrintArray1( first + 1, last );
}

void recursivePrintArray2( int first, int last )
{
   if( first < last )
      recursivePrintArray2( first, last - 1 );
   cout << " " << data[ last ];
}

void recursivePrintReverse1( int first, int last )
{



}

void recursivePrintReverse2( int first, int last )
{



}
```

```
void recursivePrintArray1( int first, int last )
{
   cout << " " << data[ first ];
   if( first < last )
      recursivePrintArray1( first + 1, last );
}

void recursivePrintArray2( int first, int last )
{
   if( first < last )
      recursivePrintArray2( first, last - 1 );
   cout << " " << data[ last ];
}

void recursivePrintReverse1( int first, int last )
{
   cout << " " << data[ last ];
   if( first < last )
      recursivePrintReverse1( first, last - 1 );
}
```

```
void recursivePrintArray1( int first, int last )
{
   cout << " " << data[ first ];
   if( first < last )
      recursivePrintArray1( first + 1, last );
}

void recursivePrintArray2( int first, int last )
{
   if( first < last )
      recursivePrintArray2( first, last - 1 );
   cout << " " << data[ last ];
}

void recursivePrintReverse1( int first, int last )
{
   cout << " " << data[ last ];
   if( first < last )
      recursivePrintReverse1( first, last - 1 );
}

void recursivePrintReverse2( int first, int last )
{
   if( first < last )
      recursivePrintReverse2( first + 1, last );
   cout << " " << data[ first ];
}
```

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
       0        2
```

0	1	2
9	6	7

```
int data[ size ];
int main()  3
{
   a2( 0, size - 1 );
}
       0        2
```

```
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
int data[ size ];
int main()  3
{
  a1( 0, size - 1 );
}     0      2

      0      2
```

0	1	2
9	6	7

```
int data[ size ];
int main()  3
{
  a2( 0, size - 1 );
}     0      2

      0      2
```

```
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
int data[ size ];
int main()  3
{
  a1( 0, size - 1 );
}
      0      2

      0      2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

9

0	1	2
9	6	7

```
int data[ size ];
int main()  3
{
  a2( 0, size - 1 );
}
      0      2

      0      2
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
int data[ size ];
int main()  3
{
  a1( 0, size - 1 );
}     0      2
```

```
9
```

```

```

0	1	2
9	6	7

```
int data[ size ];
int main()  3
{
  a2( 0, size - 1 );
}     0      2
```

```
         0     2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}                       1   2
```

```
         0     2
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];   0    1
}
```

```
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
    0      2
```

```
        0      2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
                 1    2
```

```
        1      2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

9

0	1	2
9	6	7

```
int data[ size ];
int main()  3
{
   a2( 0, size - 1 );
}
    0      2
```

```
        0      2
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}                   0    1
```

```
        0      1
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
int data[ size ];
int main() 3
{
   a1( 0, size - 1 );
}
       0     2
```

```
       0     2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
                    1    2
```

```
       1     2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

96

0	1	2
9	6	7

```
int data[ size ];
int main() 3
{
   a2( 0, size - 1 );
}
       0     2
```

```
                 0     2
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];   0    1
}
```

```
                 0     1
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
        0      2
```

96

0	1	2
9	6	7

```
int data[ size ];
int main()  3
{
   a2( 0, size - 1 );
}
        0      2
```

```
          0       2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}                      1    2
```

```
          1       2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}                      2    2
```

```
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
          0       2
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];  0    1
}
```

```
          0       1
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];  0    0
}
```

```
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}   0       2
```

```
96
```

```
int data[ size ];
int main()  3
{
   a2( 0, size - 1 );
}   0       2
```

0	1	2
9	6	7

```
                 0       2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}                       1     2

                 1       2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}                       2     2

                 2       2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
                 0       2
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];   0     1
}

                 0       1
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];   0     0
}

                 0       0
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}      0      2
```

967

9

```
int data[ size ];
int main()  3
{
   a2( 0, size - 1 );
}      0      2
```

0	1	2
9	6	7

```
                0     2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}                        1    2
```

```
                1     2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}                        2    2
```

```
                2     2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
                0     2
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];  0    1
}
```

```
                0     1
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];  0    0
}
```

```
                0     0
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
      0     2
```

```
        0     2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}                       1    2
```

```
        1     2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}                       2    2
```

```
        2     2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

967

96

0	1	2
9	6	7

```
int data[ size ];
int main()  3
{
   a2( 0, size - 1 );
}
      0     2
```

```
                  0     2
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];   0    1
}
```

```
                  0     1
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];   0    0
}
```

```
                  0     0
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
int data[ size ];
int main()  3
{
  a1( 0, size - 1 );
}     0      2
```

```
967
```

```
967
```

```
int data[ size ];
int main()  3
{
  a2( 0, size - 1 );
}     0      2
```

```
0 1 2
9 6 7
```

```
           0       2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}                         1   2
```

```
           0       2
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];  0    1
}
```

```
           1       2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}                         2   2
```

```
           0       1
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];  0    0
}
```

```
           2       2
void a1( int f, int l )
{
   cout << d[ f ];
   if( f < l ) a1( f + 1, l );
}
```

```
           0       0
void a2( int f, int l )
{
   if( f < l ) a2( f, l - 1 );
   cout << d[ l ];
}
```

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
      0       2
```

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
      0       2
```

0	1	2
9	6	7

```
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
int data[ size ];
int main()  3
{
  a1( 0, size - 1 );
}
     0       2

     0     2
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

0	1	2
9	6	7

```
int data[ size ];
int main()  3
{
  a1( 0, size - 1 );
}
     0       2

     0     2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
       0      2
```

```
7
```

0	1	2
9	6	7

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
       0      2
```

```
         0      2
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
         0      2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
int data[ size ];
int main()  3
{  a1( 0, size - 1 );
}
       0      2

       0      2
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
                  0    1
```

7

0	1	2
9	6	7

```
int data[ size ];
int main()  3
{  a1( 0, size - 1 );
}
       0      2

       0      2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];          1    2
}
```

```
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
        0     2
```

```
        0     2
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
                   0    1
```

```
        0     1
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

7

0	1	2
9	6	7

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
        0     2
```

```
             0     2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];        1    2
}
```

```
             1     2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
int data[ size ];
int main()  3
{
  a1( 0, size - 1 );
}
    0      2
```

76

```
int data[ size ];
int main()  3
{
  a1( 0, size - 1 );
}
    0      2
```

0	1	2
9	6	7

```
             0      2
void r1( int f, int l )
{
  cout << d[ l ];
  if( f < l ) r1( f, l - 1 );
}                 0     1
```

```
                  0      2
void r2( int f, int l )
{
  if( f < l ) r2( f + 1, l );
  cout << d[ f ];     1    2
}
```

```
             0      1
void r1( int f, int l )
{
  cout << d[ l ];
  if( f < l ) r1( f, l - 1 );
}
```

```
                  1      2
void r2( int f, int l )
{
  if( f < l ) r2( f + 1, l );
  cout << d[ f ];
}
```

```
void r1( int f, int l )
{
  cout << d[ l ];
  if( f < l ) r1( f, l - 1 );
}
```

```
void r2( int f, int l )
{
  if( f < l ) r2( f + 1, l );
  cout << d[ f ];
}
```

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
      0      2
```

76

0	1	2
9	6	7

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
      0      2
```

```
                0     2
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
                   0    1
```

```
                0     1
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
                   0    0
```

```
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
                0     2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];        1    2
}
```

```
                1     2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];        2    2
}
```

```
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
       0       2
```

76

0	1	2
9	6	7

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
       0       2
```

```
                 0           2
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
                   0       1
```

```
                 0           1
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
                   0       0
```

```
                 0           0
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
                 0           2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];        1    2
}
```

```
                 1           2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];        2    2
}
```

```
                 2           2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
int data[ size ];
int main()  3
{
  a1( 0, size - 1 );
}   0       2
```

769

7

```
int data[ size ];
int main()  3
{
  a1( 0, size - 1 );
}   0       2
```

0	1	2
9	6	7

```
          0      2
void r1( int f, int l )
{
  cout << d[ l ];
  if( f < l ) r1( f, l - 1 );
}                 0    1
```

```
          0      1
void r1( int f, int l )
{
  cout << d[ l ];
  if( f < l ) r1( f, l - 1 );
}                 0    0
```

```
          0      0
void r1( int f, int l )
{
  cout << d[ l ];
  if( f < l ) r1( f, l - 1 );
}
```

```
          0      2
void r2( int f, int l )
{
  if( f < l ) r2( f + 1, l );
  cout << d[ f ];       1    2
}
```

```
          1      2
void r2( int f, int l )
{
  if( f < l ) r2( f + 1, l );
  cout << d[ f ];       2    2
}
```

```
          2      2
void r2( int f, int l )
{
  if( f < l ) r2( f + 1, l );
  cout << d[ f ];
}
```

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
       0      2
```

769

76

```
int data[ size ];
int main()  3
{
   a1( 0, size - 1 );
}
       0      2
```

0	1	2
9	6	7

```
                 0      2
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
                   0    1
```

```
                 0      1
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
                   0    0
```

```
                 0      0
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
                 0      2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];          1    2
}
```

```
                 1      2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];          2    2
}
```

```
                 2      2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

```
int data[ size ];
int main()  3
{   a1( 0, size - 1 );
}       0      2
```

769

769

```
int data[ size ];
int main()  3
{   a1( 0, size - 1 );
}       0      2
```

0	1	2
9	6	7

```
                 0      2
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}                     0    1
```

```
                 0      1
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}                     0    0
```

```
                 0      0
void r1( int f, int l )
{
   cout << d[ l ];
   if( f < l ) r1( f, l - 1 );
}
```

```
                 0      2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];        1    2
}
```

```
                 1      2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];        2    2
}
```

```
                 2      2
void r2( int f, int l )
{
   if( f < l ) r2( f + 1, l );
   cout << d[ f ];
}
```

Recursive sum of elements in an array

Let $A_n = (a_0, a_1, \ldots, a_n)$.

Let $sum(A_n) = a_0 + a_1 + \cdots + a_n$.

Then $sum(A_0) = a_0$.

$$sum(A_n) = sum(A_{n-1}) + a_n.$$

```
const int arraySize = 3;
int data[ arraySize ];

int main()
{
   for( int i = 0; i < arraySize; i++ )
         data[ i ] = rand() % 10;

   cout << "The sum of elements in the array is: "
        << sum( arraySize - 1 ) << endl << endl;
}

int sum( int last )
{




}
```

$$sum(A_0) = a_0.$$

$$sum(A_n) = sum(A_{n-1}) + a_n.$$

```
const int arraySize = 3;
int data[ arraySize ];

int main()
{
   for( int i = 0; i < arraySize; i++ )
         data[ i ] = rand() % 10;

   cout << "The sum of elements in the array is: "
        << sum( arraySize - 1 ) << endl << endl;
}

int sum( int last )
{
   if( last == 0 )
      return           ;
   else
      return                                ;
}
```

$$sum(A_0) = a_0.$$

$$sum(A_n) = sum(A_{n-1}) + a_n.$$

$$sum(A_0) = a_0.$$

$$sum(A_n) = sum(A_{n-1}) + a_n.$$

```
const int arraySize = 3;
int data[ arraySize ];

int main()
{
   for( int i = 0; i < arraySize; i++ )
         data[ i ] = rand() % 10;

   cout << "The sum of elements in the array is: "
        << sum( arraySize - 1 ) << endl << endl;
}

int sum( int last )
{
   if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ];
}
```

```
const int arraySize = 1;
int data[ arraySize ];

int main()
{
   for( int i = 0; i < arraySize; i++ )
         data[ i ] = rand() % 10;

   cout << "The sum of elements in the array is: "
        << sum( arraySize - 1 ) << endl << endl;
}                     0
```

```
 int sum( int last )
 {
    if( last == 0 )
       return data[ 0 ];
    else
       return sum( last - 1 ) + data[ last ];
 }
```

```
const int arraySize = 2;
int data[ arraySize ];

int main()
{
   for( int i = 0; i < arraySize; i++ )
         data[ i ] = rand() % 10;

   cout << "The sum of elements in the array is: "
        << sum( arraySize - 1 ) << endl << endl;
}                     1
```

```
             □
int sum( int last )
{
   if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ];
}                     0
```

```
const int arraySize = 3;
int data[ arraySize ];

int main()
{
   for( int i = 0; i < arraySize; i++ )
         data[ i ] = rand() % 10;

   cout << "The sum of elements in the array is: "
        << sum( arraySize - 1 ) << endl << endl;
}                     2




               □
 int sum( int last )
 {
    if( last == 0 )
       return data[ 0 ];
    else
       return sum( last - 1 ) + data[ last ];
 }                     1
```

```
const int arraySize = 4;
int data[ arraySize ];

int main()
{
   for( int i = 0; i < arraySize; i++ )
         data[ i ] = rand() % 10;

   cout << "The sum of elements in the array is: "
        << sum( arraySize - 1 ) << endl << endl;
}                     3
```

```
             □
int sum( int last )
{
   if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ];
}                     2
```

```
int main()
                3
{  int data[ arraySize ];
   cout << sum( arraySize - 1 ) << endl << endl; }
                       2
```

```
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ]; }
```

0	1	2
9	6	7

```
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ]; }
```

```
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ]; }
```

```
int main()
                 3
{  int data[ arraySize ];
   cout << sum( arraySize - 1 ) << endl << endl; }
                       2
```

```
                2
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ]; }
                       1                7
```

0	1	2
9	6	7

```
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ]; }
```

```
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ]; }
```

```
                 3
int main()
{  int data[ arraySize ];
   cout << sum( arraySize - 1 ) << endl << endl; }
                       2
```

```
               2
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ]; }
                       1               7
```

0	1	2
9	6	7

```
               1
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ]; }
                       0               6
```

```
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ]; }
```

```
int main()
                3
{  int data[ arraySize ];
   cout << sum( arraySize - 1 ) << endl << endl; }
                       2


               2
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ]; }
                       1                7


               1
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ]; }
                       0                6


               0
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else         9
      return sum( last - 1 ) + data[ last ]; }
```

0	1	2
9	6	7

```
int main()
                3
{  int data[ arraySize ];
   cout << sum( arraySize - 1 ) << endl << endl; }
                       2



               2
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else
      return sum( last - 1 ) + data[ last ]; }
                       1                7


               1
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else                         15
      return sum( last - 1 ) + data[ last ]; }
               9       0                6


               0
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else         9
      return sum( last - 1 ) + data[ last ]; }
```

0	1	2
9	6	7

```
int main()
                3
{  int data[ arraySize ];
   cout << sum( arraySize - 1 ) << endl << endl; }
                       2

               2
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else                     22
      return sum( last - 1 ) + data[ last ]; }
              15        1               7

               1
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else                     15
      return sum( last - 1 ) + data[ last ]; }
               9        0               6

               0
int sum( int last )
{  if( last == 0 )
      return data[ 0 ];
   else          9
      return sum( last - 1 ) + data[ last ]; }
```

0	1	2
9	6	7

```
int recursiveSum1( int first, int last )
{
   if( first == last )
      return data[ last ];

   return recursiveSum1( first, last - 1 ) + data[ last ];
}

int recursiveSum2( int first, int last )
{
   if( first == last )
      return data[ first ];

   return recursiveSum2( first + 1, last ) + data[ first ];
}
```

```
int sum1( int f, int l )
{
   if( f == l ) return d[l];
   return sum1( f, l-1 ) + d[l];
}

int sum2( int f, int l )
{
   if( f == l ) return d[f];
   return sum2( f+1, l ) + d[f];
}
```

```
int data[ size ];
int main()  3
{
  cout << sum( 0, size - 1 );
}
                 0      2
```

0	1	2
9	6	7

```
int sum( int f, int l )
{
  if( f == l ) return d[ l ];
  return sum( f, l - 1 ) + d[ l ];
}
```

```
int sum( int f, int l )
{
  if( f == l ) return d[ l ];
  return sum( f, l - 1 ) + d[ l ];
}
```

```
int sum( int f, int l )
{
  if( f == l ) return d[ l ];
  return sum( f, l - 1 ) + d[ l ];
}
```

```
int data[ size ];
int main()  3
{
  cout << sum( 0, size - 1 );
}
                 0      2
```

```
int sum( int f, int l )
{
  if( f == l ) return d[ f ];
  return sum( f + 1, l ) + d[ f ];
}
```

```
int sum( int f, int l )
{
  if( f == l ) return d[ f ];
  return sum( f + 1, l ) + d[ f ];
}
```

```
int sum( int f, int l )
{
  if( f == l ) return d[ f ];
  return sum( f + 1, l ) + d[ f ];
}
```

```
int data[ size ];
int main()  3
{
  cout << sum( 0, size - 1 );
}
                0      2
```

0	1	2
9	6	7

```
           0      2
int sum( int f, int l )
{
  if( f == l ) return d[ l ];
  return sum( f, l - 1 ) + d[ l ];
}
```

```
int sum( int f, int l )
{
  if( f == l ) return d[ l ];
  return sum( f, l - 1 ) + d[ l ];
}
```

```
int sum( int f, int l )
{
  if( f == l ) return d[ l ];
  return sum( f, l - 1 ) + d[ l ];
}
```

```
int data[ size ];
int main()  3
{
  cout << sum( 0, size - 1 );
}
                0      2
```

```
           0      2
int sum( int f, int l )
{
  if( f == l ) return d[ f ];
  return sum( f + 1, l ) + d[ f ];
}
```

```
int sum( int f, int l )
{
  if( f == l ) return d[ f ];
  return sum( f + 1, l ) + d[ f ];
}
```

```
int sum( int f, int l )
{
  if( f == l ) return d[ f ];
  return sum( f + 1, l ) + d[ f ];
}
```

```
int data[ size ];
int main()  3
{
   cout << sum( 0, size - 1 );
}
                  0       2
```

0	1	2
9	6	7

```
             0       2
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l - 1 ) + d[ l ];
}
               0    1
```

```
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l - 1 ) + d[ l ];
}
```

```
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l - 1 ) + d[ l ];
}
```

```
int data[ size ];
int main()  3
{
   cout << sum( 0, size - 1 );
}
                  0       2
```

```
             0       2
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f + 1, l ) + d[ f ];
}
                  1    2
```

```
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f + 1, l ) + d[ f ];
}
```

```
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f + 1, l ) + d[ f ];
}
```

```
int data[ size ];
int main()  3
{
  cout << sum( 0, size - 1 );
}                0      2
```

0	1	2
9	6	7

```
           0      2
int sum( int f, int l )
{
  if( f == l ) return d[ l ];
  return sum( f, l-1 ) + d[ l ];
}             0   1
```

```
           0      1
int sum( int f, int l )
{
  if( f == l ) return d[ l ];
  return sum( f, l-1 ) + d[ l ];
}
```

```
int sum( int f, int l )
{
  if( f == l ) return d[ l ];
  return sum( f, l-1 ) + d[ l ];
}
```

```
int data[ size ];
int main()  3
{
  cout << sum( 0, size - 1 );
}                0      2
```

```
           0      2
int sum( int f, int l )
{
  if( f == l ) return d[ f ];
  return sum( f+1, l ) + d[ f ];
}              1    2
```

```
           1      2
int sum( int f, int l )
{
  if( f == l ) return d[ f ];
  return sum( f+1, l ) + d[ f ];
}
```

```
int sum( int f, int l )
{
  if( f == l ) return d[ f ];
  return sum( f+1, l ) + d[ f ];
}
```

```
int data[ size ];
int main()  3
{
  cout << sum( 0, size - 1 );
}
               0       2
```

0	1	2
9	6	7

```
               0       2
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l-1 ) + d[ l ];
}
               0    1
```

```
               0       1
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l-1 ) + d[ l ];
}
               0    0
```

```
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l-1 ) + d[ l ];
}
```

```
int data[ size ];
int main()  3
{
  cout << sum( 0, size - 1 );
}
               0       2
```

```
                 0       2
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
                 1    2
```

```
                 1       2
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
                 2    2
```

```
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
```

```
int data[ size ];
int main()  3
{
   cout << sum( 0, size - 1 );
}
                0       2
```

0	1	2
9	6	7

```
             0       2
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l-1 ) + d[ l ];
}
               0   1
```

```
             0       1
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l-1 ) + d[ l ];
}
               0   0
```

```
             0       0
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l-1 ) + d[ l ];
}
```

```
int data[ size ];
int main()  3
{
   cout << sum( 0, size - 1 );
}
                0       2
```

```
             0       2
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
                 1    2
```

```
             1       2
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
                 2    2
```

```
             2       2
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
```

```
int data[ size ];
int main()  3
{
  cout << sum( 0, size - 1 );
}
                 0      2
```

0	1	2
9	6	7

```
          0      2
int sum( int f, int l )
{
  if( f == l ) return d[ l ];
  return sum( f, l - 1 ) + d[ l ];
}
            0    1
```

```
          0      1
int sum( int f, int l )
{
  if( f == l ) return d[ l ];
  return sum( f, l - 1 ) + d[ l ];
}
            0    0
```

```
          0      0
int sum( int f, int l )
{                      9
  if( f == l ) return d[ l ];
  return sum( f, l - 1 ) + d[ l ];
}
```

```
int data[ size ];
int main()  3
{
  cout << sum( 0, size - 1 );
}
                 0      2
```

```
          0      2
int sum( int f, int l )
{
  if( f == l ) return d[ f ];
  return sum( f+1, l ) + d[ f ];
}
              1    2
```

```
          1      2
int sum( int f, int l )
{
  if( f == l ) return d[ f ];
  return sum( f+1, l ) + d[ f ];
}
              2    2
```

```
          2      2
int sum( int f, int l )
{                      7
  if( f == l ) return d[ f ];
  return sum( f+1, l ) + d[ f ];
}
```

```
int data[ size ];
int main()  3
{
  cout << sum( 0, size - 1 );
}
                 0      2
```

```
0 1 2
9 6 7
```

```
         0     2
int sum( int f, int l )
{
   if( f == l ) return d[l];
   return sum( f, l-1 ) + d[l];
}
               0   1
```

```
         0     1
int sum( int f, int l )
{
   if( f == l ) return d[l];
   return sum( f, l-1 ) + d[l];
}
            9  0   0
```

```
         0     0
int sum( int f, int l )
                          9
{
   if( f == l ) return d[l];
   return sum( f, l-1 ) + d[l];
}
```

```
int data[ size ];
int main()  3
{
  cout << sum( 0, size - 1 );
}
                 0      2
```

```
         0     2
int sum( int f, int l )
{
   if( f == l ) return d[f];
   return sum( f+1, l ) + d[f];
}
                 1   2
```

```
         1     2
int sum( int f, int l )
{
   if( f == l ) return d[f];
   return sum( f+1, l ) + d[f];
}
             7   2   2
```

```
         2     2
int sum( int f, int l )
                          7
{
   if( f == l ) return d[f];
   return sum( f+1, l ) + d[f];
}
```

```
int data[ size ];
int main()  3
{
   cout << sum( 0, size - 1 );
}
                0      2
```

0	1	2
9	6	7

```
               0      2
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l - 1 ) + d[ l ];
}
                  0    1

               0      1
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l - 1 ) + d[ l ];
}
              9   0    0     15

               0      0
int sum( int f, int l )
{                         9
   if( f == l ) return d[ l ];
   return sum( f, l - 1 ) + d[ l ];
}
```

```
int data[ size ];
int main()  3
{
   cout << sum( 0, size - 1 );
}
                0      2
```

```
               0      2
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
                    1    2

               1      2
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
               7    2    2  13

               2      2
int sum( int f, int l )
{                         7
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
```

```
  int data[ size ];
  int main()  3
  {
     cout << sum( 0, size - 1 );
  }
                  0      2
```

0	1	2
9	6	7

```
            0       2
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l-1 ) + d[ l ];
}
          15  0    1

            0       1
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l-1 ) + d[ l ];
}
          9   0    0     15

            0       0
int sum( int f, int l )
{                            9
   if( f == l ) return d[ l ];
   return sum( f, l-1 ) + d[ l ];
}
```

```
  int data[ size ];
  int main()  3
  {
     cout << sum( 0, size - 1 );
  }
                  0      2

            0       2
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
          13    1    2

            1       2
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
          7     2    2   13

            2       2
int sum( int f, int l )
{                            7
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
```

```
int data[ size ];
int main()  3
{
  cout << sum( 0, size - 1 );
}
                0      2
```

```
 0 1 2
|9|6|7|
```

```
             0      2
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l - 1 ) + d[ l ];
}
           15   0    1    22
```

```
             0      1
int sum( int f, int l )
{
   if( f == l ) return d[ l ];
   return sum( f, l - 1 ) + d[ l ];
}
            9   0    0    15
```

```
             0      0
int sum( int f, int l )
                          9
{
   if( f == l ) return d[ l ];
   return sum( f, l - 1 ) + d[ l ];
}
```

```
int data[ size ];
int main()  3
{
  cout << sum( 0, size - 1 );
}
                0      2
```

```
             0      2
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
           13    1    2   22
```

```
             1      2
int sum( int f, int l )
{
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
            7    2    2   13
```

```
             2      2
int sum( int f, int l )
                          7
{
   if( f == l ) return d[ f ];
   return sum( f+1, l ) + d[ f ];
}
```

```
int data[ size ];
int main()  3
{
   cout << sum( 0, size - 1 );
}
           22   0      2
```

0	1	2
9	6	7

```
             0      2
int sum( int f, int l )
{
   if( f == l ) return d[l];
   return sum( f, l-1 ) + d[l];
}
           15   0   1   22
```

```
             0      1
int sum( int f, int l )
{
   if( f == l ) return d[l];
   return sum( f, l-1 ) + d[l];
}
            9   0   0   15
```

```
             0      0
int sum( int f, int l )
                         9
{
   if( f == l ) return d[l];
   return sum( f, l-1 ) + d[l];
}
```

```
int data[ size ];
int main()  3
{
   cout << sum( 0, size - 1 );
}
           22   0      2
```

```
             0      2
int sum( int f, int l )
{
   if( f == l ) return d[f];
   return sum( f+1, l ) + d[f];
}
           13     1   2   22
```

```
             1      2
int sum( int f, int l )
{
   if( f == l ) return d[f];
   return sum( f+1, l ) + d[f];
}
            7     2   2   13
```

```
             2      2
int sum( int f, int l )
                         7
{
   if( f == l ) return d[f];
   return sum( f+1, l ) + d[f];
}
```

Recursive maximum of elements in an array

Let $A_n = (a_0, a_1, \ldots, a_n)$.

Let $max(A_n) = max\{a_0, a_1, \ldots, a_n\}$.

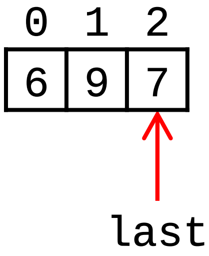
Then $max(A_0) = a_0$.

$$max(A_n) = max\{max(A_{n-1}), a_n\}.$$

```
const int size = 3;
int data[ size ];

int main()
{
   for( int i = 0; i < size; i++ )
      data[ i ] = rand() % 10;

   cout << maximum( size - 1 );
}

int maximum( int last )
{








}
```

$$max(A_0) = a_0.$$

$$max(A_n) = max\{max(A_{n-1}), a_n\}.$$



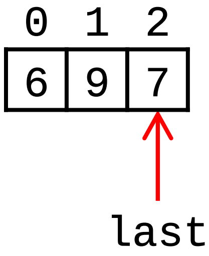
0	1	2
6	9	7

last

```
const int size = 3;
int data[ size ];

int main()
{
   for( int i = 0; i < size; i++ )
      data[ i ] = rand() % 10;

   cout << maximum( size - 1 );
}

int maximum( int last )
{
   if( last == 0 )
       return data[ 0 ];







   return max;
}
```

$$max(A_0) = a_0.$$

$$max(A_n) = max\{max(A_{n-1}), a_n\}.$$



0 1 2

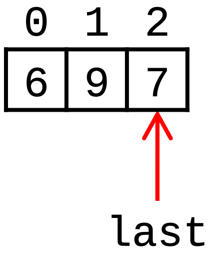
6 9 7

last

```
const int size = 3;
int data[ size ];

int main()
{
   for( int i = 0; i < size; i++ )
      data[ i ] = rand() % 10;

   cout << maximum( size - 1 );
}

int maximum( int last )
{
   if( last == 0 )
       return data[ 0 ];

   int max = maximum( last - 1 );




   return max;
}
```

$$max(A_0) = a_0.$$

$$max(A_n) = max\{max(A_{n-1}), a_n\}.$$



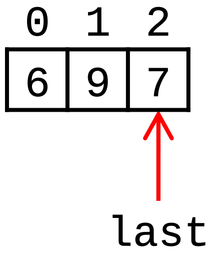
0	1	2
6	9	7

last

```
const int size = 3;
int data[ size ];

int main()
{
   for( int i = 0; i < size; i++ )
      data[ i ] = rand() % 10;

   cout << maximum( size - 1 );
}

int maximum( int last )
{
   if( last == 0 )
       return data[ 0 ];

   int max = maximum( last - 1 );

   if( max < data[ last ] )
       max = data[ last ];

   return max;
}
```

$$max(A_0) = a_0.$$

$$max(A_n) = max\{max(A_{n-1}), a_n\}.$$



```
int data[ size ];
int main()  3
{                       2
   cout << maximum( size - 1 ) << endl;
}


int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}

int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}

int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
```

0	1	2
6	9	7

```
int data[ size ];
int main()  3
{                       2
   cout << maximum( size - 1 ) << endl;
}


                   2
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}

int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}

int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
```

0	1	2
6	9	7

```
int data[ size ];
int main()  3
{                                   2
   cout << maximum( size - 1 ) << endl;
}


                         2
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );
                            1

   if( max < data[ last ] ) max = data[ last ];
   return max;
}

int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}

int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
```

0	1	2
6	9	7

```
int data[ size ];
int main()  3
                        2
{
   cout << maximum( size - 1 ) << endl;
}

                   2
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );
                          1

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
                   1
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}

int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
```

0	1	2
6	9	7

```
int data[ size ];
int main()  3
{                      2
  cout << maximum( size - 1 ) << endl;
}
```

```
                   2
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );
                          1

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
```

```
                   1
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );
                          0

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
```

```
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
```

0	1	2
6	9	7

```
int data[ size ];
int main()  3
{                       2
  cout << maximum( size - 1 ) << endl;
}
```

```
              2
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );
                          1

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
              1
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );
                          0

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
              0
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
```

0	1	2
6	9	7

```
int data[ size ];
int main()  3
{                       2
   cout << maximum( size - 1 ) << endl;
}


                   2
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );
                           1

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
                   1
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );
                 6         0

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
                   0
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
```

0	1	2
6	9	7

```
int data[ size ];
int main()  3
                        2
{
   cout << maximum( size - 1 ) << endl;
}
```

```
                   2
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );
                 9         1

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
                   1
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );
                 6         0

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
                   0
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
```

0	1	2
6	9	7

```
int data[ size ];
int main()  3
{                        2
   cout << maximum( size - 1 ) << endl;
}             9

                   2
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );
                 9         1

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
                   1
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );
                 6         0

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
                   0
int maximum( int last )
{
   if( last == 0 ) return data[ 0 ];
   int max = maximum( last - 1 );

   if( max < data[ last ] ) max = data[ last ];
   return max;
}
```

0	1	2
6	9	7

```
int recursiveMaximum1( int first, int last )
{

}

int recursiveMaximum2( int first, int last )
{

}
```

```
int recursiveMaximum1( int first, int last )
{
   if( first == last )
       return data[ last ];





   return max;
}

int recursiveMaximum2( int first, int last )
{
   if( first == last )
       return data[ first ];





   return max;
}
```

```
int recursiveMaximum1( int first, int last )
{
   if( first == last )
       return data[ last ];

   int max = recursiveMaximum1( first, last - 1 );
   if( max < data[ last ] )
       max = data[ last ];

   return max;
}

int recursiveMaximum2( int first, int last )
{
   if( first == last )
       return data[ first ];

   int max = recursiveMaximum2( first + 1, last );
   if( max < data[ first ] )
       max = data[ first ];

   return max;
}
```

```
                  0      2
int maximum1( int f, int l )
{
   if( f == l ) return d[l];
   int m = maximum1( f, l-1 );
                     0   1

   if( m < d[l] ) m = d[l];
   return m;
}
```

```
                  0      2
int maximum2( int f, int l )
{
   if( f == l ) return d[f];
   int m = maximum2( f+1, l );
                      1   2

   if( m < d[f] ) m = d[f];
   return m;
}
```

0	1	2
6	9	7

```
int maximum1( int f, int l )
{
   if( f == l ) return d[l];
   int m = maximum1( f, l-1 );

   if( m < d[l] ) m = d[l];
   return m;
}
```

```
int maximum2( int f, int l )
{
   if( f == l ) return d[f];
   int m = maximum2( f+1, l );

   if( m < d[f] ) m = d[f];
   return m;
}
```

```
int maximum1( int f, int l )
{
   if( f == l ) return d[l];
   int m = maximum1( f, l-1 );
   if( m < d[l] ) m = d[l];
   return m;
}
```

```
int maximum2( int f, int l )
{
   if( f == l ) return d[f];
   int m = maximum2( f+1, l );
   if( m < d[f] ) m = d[f];
   return m;
}
```

```
                  0      2
int maximum1( int f, int l )
{
   if( f == l ) return d[l];
   int m = maximum1( f, l-1 );
                     0   1

   if( m < d[l] ) m = d[l];
   return m;
}
```

```
                  0      1
int maximum1( int f, int l )
{
   if( f == l ) return d[l];
   int m = maximum1( f, l-1 );

   if( m < d[l] ) m = d[l];
   return m;
}
```

```
int maximum1( int f, int l )
{
   if( f == l ) return d[l];
   int m = maximum1( f, l-1 );
   if( m < d[l] ) m = d[l];
   return m;
}
```

0	1	2
6	9	7

```
                  0      2
int maximum2( int f, int l )
{
   if( f == l ) return d[f];
   int m = maximum2( f+1, l );
                       1  2

   if( m < d[f] ) m = d[f];
   return m;
}
```

```
                  1      2
int maximum2( int f, int l )
{
   if( f == l ) return d[f];
   int m = maximum2( f+1, l );

   if( m < d[f] ) m = d[f];
   return m;
}
```

```
int maximum2( int f, int l )
{
   if( f == l ) return d[f];
   int m = maximum2( f+1, l );
   if( m < d[f] ) m = d[f];
   return m;
}
```

```
                  0      2
int maximum1( int f, int l )
{
   if( f == l ) return d[l];
   int m = maximum1( f, l-1 );
                     0  1

   if( m < d[l] ) m = d[l];
   return m;
}
```

```
                  0      1
int maximum1( int f, int l )
{
   if( f == l ) return d[l];
   int m = maximum1( f, l-1 );
                     0  0

   if( m < d[l] ) m = d[l];
   return m;
}
```

```
int maximum1( int f, int l )
{
   if( f == l ) return d[l];
   int m = maximum1( f, l-1 );
   if( m < d[l] ) m = d[l];
   return m;
}
```

0	1	2
6	9	7

```
                  0      2
int maximum2( int f, int l )
{
   if( f == l ) return d[f];
   int m = maximum2( f+1, l );
                     1    2

   if( m < d[f] ) m = d[f];
   return m;
}
```

```
                  1      2
int maximum2( int f, int l )
{
   if( f == l ) return d[f];
   int m = maximum2( f+1, l );
                     2    2

   if( m < d[f] ) m = d[f];
   return m;
}
```

```
int maximum2( int f, int l )
{
   if( f == l ) return d[f];
   int m = maximum2( f+1, l );
   if( m < d[f] ) m = d[f];
   return m;
}
```

```
                  0      2
int maximum1( int f, int l )
{
   if( f == l ) return d[l];
   int m = maximum1( f, l-1 );
                     0  1

   if( m < d[l] ) m = d[l];
   return m;
}
```

```
                  0      1
int maximum1( int f, int l )
{
   if( f == l ) return d[l];
   int m = maximum1( f, l-1 );
                     0  0

   if( m < d[l] ) m = d[l];
   return m;
}
```

```
                  0      0
int maximum1( int f, int l )
{
   if( f == l ) return d[l];
   int m = maximum1( f, l-1 );
   if( m < d[l] ) m = d[l];
   return m;
}
```

0	1	2
6	9	7

```
                  0      2
int maximum2( int f, int l )
{
   if( f == l ) return d[f];
   int m = maximum2( f+1, l );
                      1   2

   if( m < d[f] ) m = d[f];
   return m;
}
```

```
                  1      2
int maximum2( int f, int l )
{
   if( f == l ) return d[f];
   int m = maximum2( f+1, l );
                      2   2

   if( m < d[f] ) m = d[f];
   return m;
}
```

```
                  2      2
int maximum2( int f, int l )
{
   if( f == l ) return d[f];
   int m = maximum2( f+1, l );
   if( m < d[f] ) m = d[f];
   return m;
}
```

```
                  0      2
int maximum1( int f, int l )
{
   if( f == l ) return d[ l ];
   int m = maximum1( f, l-1 );
               9     0  1

   if( m < d[ l ] ) m = d[ l ];
   return m;
}
```

```
                  0      1
int maximum1( int f, int l )
{
   if( f == l ) return d[ l ];
   int m = maximum1( f, l-1 );
               6     0  0

   if( m < d[ l ] ) m = d[ l ];
   return m;
}
```

```
                  0      0
int maximum1( int f, int l )
{
   if( f == l ) return d[ l ];
   int m = maximum1( f, l-1 );
   if( m < d[ l ] ) m = d[ l ];
   return m;
}
```

0	1	2
6	9	7

```
                  0      2
int maximum2( int f, int l )
{
   if( f == l ) return d[ f ];
   int m = maximum2( f+1, l );
               9      1   2

   if( m < d[ f ] ) m = d[ f ];
   return m;
}
```

```
                  1      2
int maximum2( int f, int l )
{
   if( f == l ) return d[ f ];
   int m = maximum2( f+1, l );
               7      2   2

   if( m < d[ f ] ) m = d[ f ];
   return m;
}
```

```
                  2      2
int maximum2( int f, int l )
{
   if( f == l ) return d[ f ];
   int m = maximum2( f+1, l );
   if( m < d[ f ] ) m = d[ f ];
   return m;
}
```

```
                  0      2
int maximum1( int f, int l )
{  if( f == l ) return d[ l ];
   int m = maximum1( f, l-1 );
               9     0   1

   if( m < d[ l ] ) m = d[ l ];
   return m;
}         9
```

```
                  0      1
int maximum1( int f, int l )
{  if( f == l ) return d[ l ];
   int m = maximum1( f, l-1 );
               6     0   0

   if( m < d[ l ] ) m = d[ l ];
   return m;
}
```

```
                  0      0
int maximum1( int f, int l )
{  if( f == l ) return d[ l ];
   int m = maximum1( f, l-1 );
   if( m < d[ l ] ) m = d[ l ];
   return m;
}
```

0	1	2
6	9	7

```
                  0      2
int maximum2( int f, int l )
{  if( f == l ) return d[ f ];
   int m = maximum2( f+1, l );
               9      1   2

   if( m < d[ f ] ) m = d[ f ];
   return m;
}         9
```

```
                  1      2
int maximum2( int f, int l )
{  if( f == l ) return d[ f ];
   int m = maximum2( f+1, l );
               7      2   2

   if( m < d[ f ] ) m = d[ f ];
   return m;
}
```

```
                  2      2
int maximum2( int f, int l )
{  if( f == l ) return d[ f ];
   int m = maximum2( f+1, l );
   if( m < d[ f ] ) m = d[ f ];
   return m;
}
```

Recursive greatest common divisor

```
int main()
{
   int x;
   int y;

   cout << "Enter two integers: ";
   cin >> x >> y;

   cout << "Greatest common divisor of " << x << " and "
        << y << " is " << gcd( x, y ) << endl;
}

int gcd( int a, int b )
{



}
```

```
int main()
{
   int x;
   int y;

   cout << "Enter two integers: ";
   cin >> x >> y;

   cout << "Greatest common divisor of " << x << " and "
        << y << " is " << gcd( x, y ) << endl;
}

int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}
```

```
             30     18
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}
```

```
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}
```

```
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}
```

```
            30     18
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}             18    12
```

```
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}
```

```
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}
```

```
            30      18
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}             18    12
```

```
            18      12
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}
```

```
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}
```

```
             30     18
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}              18   12
```

```
             18     12
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}              12   6
```

```
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}
```

```
            30     18
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}             18    12
```

```
            18     12
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}             12     6
```

```
            12      6
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}
```

```
            30      18
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}          6   18   12
```

```
            18      12
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}          6   12   6
```

```
            12      6
int gcd( int a, int b )
{
   if( a % b == 0 )
      return b;
   return gcd( b, a % b );
}
```